

برنامه نویسی صفحات وب

توسط: علیرضا جمیری

Jomeiri@yahoo.com

آشنایی با CSS

- روشی است برای قالب بندی و طراحی اجزای صفحه.

- مزایای CSS:

- طراحی ظاهر صفحات بدون استفاده از دستورات HTML
- جلوگیری از تکرار دستورات
- سرعت بارگذاری بیشتر برای صفحات
- استفاده از فایل CSS خارجی جهت تغییر ظاهر هر تعداد صفحه ای که مایل هستیم.

طریقه نوشتن کدهای CSS

- برای نوشتن دستورات CSS باید مفاهیم زیر را بشناسیم:
 - Selector
 - Property
 - Value
- هر دستور CSS شامل این سه قسمت است. مثال:

`H1 {color: blue}`

- یعنی تمامی متونی که با تگ `h1` نوشته می شوند به رنگ آبی باشند. در این دستور `h1` همان selector است. عبارت `color` همان property بوده و `blue` نیز value می باشد.

آشنایی با دو شناسه بسیار پرکاربرد class و id

- اغلب تگهای html دارای این دو شناسه می باشند. فرض کنید می خواهیم در صفحه دو نوع پاراگراف داشته باشیم. یکی با متن آبی و دیگری با رنگ قرمز. برای مشخص نمودن این دو نوع پاراگراف از شناسه class استفاده می کنیم:

```
P.blue {color: blue}
```

```
P.red {color: red}
```

- در ادامه:

```
<p class="blue"> Alireza Jomeiri </p>
```

```
<p class="red"> Alireza Jomeiri </p>
```

آشنایی با دو شناسه بسیار پرکاربرد class و id ...

- اگر بخواهیم از دو کلاس در تگ خود استفاده کنیم بصورت زیر عمل می‌کنیم:

```
P.blue {color: blue}
```

```
P.red {color: red}
```

```
.center {text-align: center}
```

```
<p class="blue center"> Alireza Jomeiri </p>
```

```
<p class="red center"> Alireza Jomeiri </p>
```

- اگر بخواهیم برای تگهایی که از شناسه id استفاده می‌کنند، یک سبک بنویسیم:

```
#border {border: green solid}
```

با این دستور در تمام صفحه تگهایی که id آنها دارای مقدار border است با کادر سبز رنگ نمایش داده می‌شوند.

دستورات CSS را کجا بنویسیم؟

- 1- شیوه نامه درون خطی (inline)
- 2- شیوه نامه داخلی (internal)
- 3- شیوه نامه خارجی (external)

1- شیوه نامه درون خطی (inline)

```
<body>
```

```
<p style="color: green"> Alireza Jomeiri</p>
```

```
</body>
```

برای تعریف بیش از یک نوع سبک از علامت ; استفاده می کنیم:

```
<body>
```

```
<p style="color: green ; font-style: italic"> Alireza Jomeiri</p>
```

```
</body>
```

2- شیوه نامه داخلی (internal)

این روش زمانی استفاده می شود که قصد دارید یک سری سبکها را فقط بر روی یک صفحه خاص اعمال کنید نه در کل صفحات.

```
<html>
<head>
  <style>
    h1 {color:green ; font-style:italic}
  </style>
</head>
<body>
  <h1>Alireza Jomeiri</h1>
</body>
</html>
```


3- شیوه نامه خارجی (external)

- در این روش تمامی دستورات CSS در یک فایل مجزا نوشته می شوند و با فرمت CSS ذخیره می شود. سپس در هر صفحه ای که می خواهیم این دستورات بر عناصر آن اعمال شوند، توسط تگ `<link>` فایل CSS خود را به صفحه متصل می کنیم.
- توجه شود که فایل های CSS فقط حاوی دستورات CSS هستند و هیچ تگ HTML نباید در آن نوشته شود.

مثلا دستورات زیر را در یک فایل با نام `style1.css` نوشته و ذخیره می کنیم:

```
h1 {color:green ; font-style:italic}
```

```
h2 {color:red ; font-style:italic}
```

3- شیوه نامه خارجی (external) ...

حال در فایل html خود باید برای استفاده از فایل style1.css بصورت زیر عمل کنیم:

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="Style1.css">
  </head>
  <body>
    <h1>Alireza Jomeiri</h1>
    <h2>Alireza Jomeiri </h2>
  </body>
</html>
```

تنظیم حاشیه، کادر و فاصله



تنظیم حاشیه، کادر و فاصله ...

خاصیت Border باعث ترسیم یک کادر به دور عنصر مورد نظر می گردد که خود به سه بخش تقسیم می شود:

1. Border-width: پهنای کادر را مشخص می کند.
2. Border-color: رنگ کادر را مشخص می کند.
3. Border-style: سبک کادر را مشخص می کند.

تنظیم حاشیه، کادر و فاصله ...

- از خاصیت `border-width` برای مشخص کردن پهنا یا عرض کادر یک عنصر استفاده می شود که یکی از مقادیر زیر را میگیرد:
 - Thin باریک، برابر با `1px`
 - Medium متوسط برابر با `3px`
 - Thick پهن، برابر با `5px`
 - مقدار عددی دلخواه برحسب یکی از واحدهای محاسباتی که معمولاً `px` است.
 - و مقدار پیشفرض این خصوصیت `medium` می باشد.
- مثال:

```
H1 {  
border-width: 2px 5px 7px 9px;  
}
```

تنظیم حاشیه، کادر و فاصله ...

- خاصیت border-color برای مشخص کردن رنگ کادر بکار می رود.
- می تواند مقادیر نام رنگ یا کد رنگ را بپذیرد.

مثال:

```
H1 {  
border-width: 2px 5px 7px 9px;  
border-color: black green red yellow;  
}
```

تنظیم حاشیه، کادر و فاصله ...

- خاصیت border-style بمنظور تعیین سبک کادر بکار می رود.
- یکی از مقادیر زیر را می گیرد:

Solid- dashed- dotted- double- groove- hidden- inset- outset- ridge

مثال:

```
H1 {  
border-width: 2px 5px 7px 9px;  
border-color: black green red yellow;  
border-style: solid dashed dotted double;  
}
```

تنظیم حاشیه، کادر و فاصله ...

کوتاه نویسی خاصیت border

- ترتیب مقداردهی به این خاصیت مهم است و جابجایی باعث از کار افتادن می شود.

`Border: border-width border-style border-color;`

مثال:

```
H1 {  
Border: 3px outset red;  
}
```

اگر بخواهیم فقط خصوصیات یک طرف کادر را تعیین کنیم از خواص زیر نیز می توانیم کمک بگیریم:

`Border-top, border-bottom, border-right, border-left`

تنظیم حاشیه، کادر و فاصله ...

- حاشیه (margin)

از این صفت بمنظور تنظیم فاصله کادر از چهار حاشیه صفحه یا عنصر دیگر استفاده می شود. یکی از سه مقدار **عددی**، **درصدی** یا **auto** را می پذیرد. مقدار پیش فرض **auto** می باشد.

چهار حالت دارد:

margin-left -1

margin-right -2

margin-top -3

margin-bottom -4

مثال:

```
<div style="margin-left:200px; border:double"> Jomeiri </div>
```

تنظیم حاشیه، کادر و فاصله ...

کوتاه نویسی خاصیت margin

چهار مقدار در جهت عقربه های ساعت تخصیص می یابند یعنی به ترتیب:

Top ، right ، bottom و left

مثال:

```
Body {margin: 70px 100px 50px 20px}
```

توجه: هنگام استفاده از دو خاصیت برای کوتاه نویسی، مقدار اول bottom و top را و مقدار دوم left و right را مشخص می کند. مثال:

```
P {margin: 2em 5em}
```

تنظیم حاشیه، کادر و فاصله ...

- فاصله (padding)

از این صفت بمنظور تنظیم فاصله یک عنصر از کادر دور آن استفاده می شود.

چهار حالت دارد:

padding-left -1

padding-right -2

padding-top -3

padding-bottom -4

مثال:

```
<div style="padding-right:100px; border-style:double"> Jomeiri </div>
```

تنظیم حاشیه، کادر و فاصله ...

- استفاده از مقادیر منفی برای padding مجاز نیست.
 - گذاشتن کلمه کلیدی auto هیچ تاثیری ندارد.
 - در padding تداخل وجود ندارد فقط در margin اتفاق می افتد.
 - می توان از فرم کوتاه نویسی padding نیز استفاده نمود.
- مثال:

```
P {  
  Padding: 3px 5px 2px 7px;  
}
```

تنظیم حاشیه، کادر و فاصله ...

- خاصیت width

عرض محتوای یک عنصر را تعیین می کند.

عرض کلی یک عنصر = عرض محتوا + عرض padding + عرض border

از این خاصیت می توان برای تخصیص یک مقدار کلی به برخی عناصر input مثل دکمه های submit و reset استفاده کرد.

مثال:

```
Input.button {  
Width: 10em;  
}
```

تنظیم حاشیه، کادر و فاصله ...

- خاصیت height

ارتفاع محتوای یک عنصر را تعیین می کند.

همانند خاصیت width از این خاصیت می توان برای مشخص کردن مثلا ابعاد یک تصویر استفاده کرد.

مثال:

```
img.footer {  
Width: 440px;  
Height: 130px;  
}
```

تنظیم حاشیه، کادر و فاصله ...

• خاصیت min-height و min-width

این دو خاصیت محدودیت کوچکترین اندازه را برای یک عنصر، تعیین می کنند و از منقبض شدن عرض یا ارتفاع یک عنصر هنگام تغییر اندازه پنجره مرورگر یا رزولوشن جلوگیری می کنند.

مقادیر مورد قبول: عدد، درصد، auto

مثال:

```
Body {  
min-width: 1024px; min-height: 768px;  
}
```

تنظیم حاشیه، کادر و فاصله ...

• خاصیت max-height و max-width

این دو خاصیت محدودیت بزرگترین اندازه را برای یک عنصر، تعیین می کنند و از منبسط شدن عرض یا ارتفاع یک عنصر هنگام تغییر اندازه پنجره مرورگر یا رزولوشن جلوگیری می کنند.

مقادیر مورد قبول: عدد، درصد، auto

مثال:

```
Body {  
max-width: 1440; max-height: auto;  
}
```


تنظیم حاشیه، کادر و فاصله ...

خاصیت overflow

- این خصوصیت هنگامی استفاده می شود که محتویات یک عنصر به دلیل محدودیت در ابعاد تعریف شده برای آن، امکان سرریز دارد، خاصیت overflow نحوه ی برخورد با این سرریز را مشخص می کند.
- به طور کلی چند مقدار را می توانیم برای پارامتر overflow استفاده کنیم:
 - **visible**
با قرار دادن این مقدار همه محتویاتی که طول یا عرض بیش از اندازه داشته اند نمایش داده خواهند شد.
 - **hidden**
باعث خواهد شد تا محتویات با طول یا عرض بیش از اندازه، نمایش داده نشوند.
 - **scroll**
محتوای سرریز شده را پنهان می کند و قابلیت اسکرول شدن برای دیدن محتوای سرریز شده را فعال می کند.
 - **Auto**
اگر زیاد بودن عرض عنصر باعث سرریز شود، اسکرول افقی و اگر زیاد بودن ارتفاع عنصر باعث سرریز شود اسکرول عمودی را برای عنصر فعال می کند.

تنظیم حاشیه، کادر و فاصله ...

مثال برای خاصیت `overflow`:

```
div {  
border: 1px solid #000;  
width: 400px;  
height: 140px;  
overflow: auto;  
}
```

```
<div>  
<h3> CSS and Web Design Tutorials </h3>  
<h3> CSS and Web Design Tutorials </h3>  
<h3> CSS and Web Design Tutorials </h3>  
<h3> CSS and Web Design Tutorials </h3>  
<h3> CSS and Web Design Tutorials </h3>  
<h3> CSS and Web Design Tutorials </h3>  
</div>
```

ویژگیهای متن در CSS

با کمک این ویژگیها می توانید نحوه قرار گرفتن متن در صفحه، فاصله بین حروف و بسیاری از خواص یک متن را تغییر دهید.

- ویژگی `letter-spacing`

فاصله بین کاراکترهای متن را تنظیم می کند.

```
<html>
```

```
<body>
```

```
<P style="letter-spacing:10"> This is a test.</P>
```

```
</body>
```

```
</html>
```

ویژگیهای متن در CSS ...

- ویژگی text-align

نحوه چینش متن را مشخص می کند. دارای مقادیر left، right، justify و center می باشد.

```
<html>
```

```
<body>
```

```
<P style="text-align:center"> This is a sample text. </P>
```

```
</body>
```

```
</html>
```

ویژگیهای متن در CSS ...

- ویژگی text-decoration

می توان برای قسمت خاصی از متن یکی از خواص زیر را اعمال نمود:

None

Underline

Overline

Line-through

Blink

- مثال: ایجاد لینک بدون زیرخط

```
<a href="#example" style="text-decoration:none"> simple text </a>
```

ویژگیهای متن در CSS ...

- ویژگی text-transform

حالت نوشتاری حروف متن را مشخص می کند. با خاصیت‌های:

None

Lowercase

Uppercase

Capitalize

- مثال:

```
<html>  
<body>  
<P style="text-transform:capitalize"> This is a test.</P>  
</body>  
</html>
```

ویژگیهای متن در CSS ...

- ویژگی text-indent مقدار تورفتگی سطر اول را مشخص می کند.

```
<html>
<body>
<div style="text-indent:20">
this is first line. <br>
This is second line.
</div>
</body>
</html>
```

ویژگیهای متن در CSS ...

ویژگی line-height

با این ویژگی ارتفاع سطرها را مشخص می کنیم.

مثال:

```
<div style="line-height: 50px"> this is first line. <br> this is second line.
</div>
```


تنظیم فونتها

ویژگی font-family

نوع فونت متن را مشخص می کند.

مثال:

```
<html>  
<body>  
<p style="font-family:tahoma"> sample of font</p>  
</body>  
</html>
```

توجه: اگر نام فونت شامل فاصله باشد باید این نام داخل جفت کوتیشن قرار بگیرد.

تنظیم فونتها...

ویژگی font-size

اندازه فونت متن را مشخص می کند.

مثال:

```
<html>  
<body>  
<p style="font-size:24"> sample of font</p>  
</body>  
</html>
```

تنظیم فونتها...

ویژگی font-style

سبک فونت متن را مشخص می کند. (oblique و italic ، normal)

مثال:

```
<html>  
<body>  
<p style="font-style:italic"> sample of font</p>  
</body>  
</html>
```

تنظیم فونتها...

ویژگی font-weight

میزان ضخامت متن را مشخص می کند.

(normal, bold, bolder, lighter, 100-900)

مثال:

```
<html>  
<body>  
<p style="font-weight:700"> sample of font</p>  
</body>  
</html>
```

تغییر رنگ متن (color)

از ویژگی **color** برای تغییر رنگ قسمت‌های مختلف صفحه استفاده می‌شود.
مثال:

```
<p style="color:red"> this is a test.</p>
```

برای تغییر رنگ بخشی از متن از تگ **span** :
مثال:

```
<p style="color:red"> Css <span style="color:blue"> is </span> Good  
</p>
```

تغییر رنگ زمینه

از ویژگی **background-color** بمنظور تعیین رنگ زمینه برای تقریبا هر چیزی در صفحه استفاده می شود.

مثال:

```
<div style="color:white;background-color:red;width:160"> this is test.  
</div>
```

نکته: مقدار transparent مشخص می کند که هیچ رنگی نباید استفاده شود.

تغییر تصویر زمینه

از ویژگی **background-image** بمنظور تعیین تصویر زمینه برای تقریبا هر چیزی در صفحه استفاده می شود.

مثال:

```
<body style="background-image:url(OurPic.jpg)"> </body>
```

مثال 2: تعیین تصویر زمینه خانه های جدول:

```
<table border="1" style="background-image: url(example1.jpg)">  
<td>first cell </td>  
<td>second cell </td>  
</table>
```

ویژگیهای تصویر زمینه

• ویژگی background-repeat

این ویژگی تعیین کننده نوع تکرار شدن تصویر زمینه است. مقادیر:

Repeat : تکرار تصویر هم بصورت عمودی و هم افقی

No repeat : تصویر زمینه نباید تکرار شود.

Repeat-x : تکرار تصویر زمینه بصورت افقی

Repeat-y : تکرار تصویر زمینه بصورت عمودی

مثال:

```
<body style="background-image:url(OurPic.jpg);  
background-repeat:repeat-y">  
<p> this is test. </p>  
</body>
```


ویژگیهای تصویر زمینه ...

• ویژگی background-attachment

تعیین موقعیت تصویر زمینه هنگام اسکرول صفحه.

مقادیر:

Scroll : تصویر هنگام اسکرول صفحه حرکت می کند.

Fixed : تصویر هنگام اسکرول صفحه ثابت می ماند. (مثلا لوگوی سایت)

مثال:

```
<body style="background-image:url(OurPic.jpg);background-attachment:fixed">
```

```
<p> Hello world...</p>
```

```
</body>
```

ویژگیهای تصویر زمینه ...

• ویژگی background-position

جهت تعیین موقعیت تصویر زمینه در صفحه.

زمانی استفاده می شود که برای تصویر background-repeat: no-repeat باشد.

مقادیر:

Top, center, bottom, left, right

می تواند دو مقدار را به خود اختصاص دهد. مثال:

Background-position: right center

و یا بجای گزینه ها از مقیاس پیکسل استفاده می شود: (40 پیکسل از چپ و 20 از بالا)

background-position: 40px 20px

و یا:

Background-position: 50% -2cm

ویژگیهای تصویر زمینه (کوتاه نویسی)

- پنج خاصیت مربوط به background را می توان بصورت کوتاه در یک دستور ادغام نمود.

- مثال:

```
table { background: #0c0 url(./images/sky.png) no-repeat fixed  
bottom right; }
```

- اگر مقداری برای خاصیت ذکر نشود، مقدار پیش فرض آن استفاده خواهد شد.

اعمال تغییر بر روی لینکها

- با استفاده از CSS می توانیم نحوه نمایش دادن لینکها (رنگ، اندازه و ...) را در صفحه تنظیم کنیم.

- برای کار با لینکها در CSS می توانیم از مشخصات آنها استفاده کنیم:

- **a:link**

این مشخصه ها نحوه نمایش لینکهایی را که در صفحه وجود دارند و هنوز روی آنها کلیک نشده است تنظیم می کنند.

- **a:hover**

این مشخصه وضعیت لینک را وقتی که نشانگر ماوس روی آن قرار دارن تنظیم می کند.

- **a:active**

این قسمت مشخص کننده وضعیت لینکی است که بر روی آن کلیک می شود.

- **a:visited**

این مشخصه برای لینکهایی که در بازدیدهای قبلی مورد استفاده قرار گرفته اند استفاده می شود.

اعمال تغییر بر روی لینکها...

• مثال

```
<style type="text/css">
<!--
#pre a:link, #pre a { color:green; font-size:12pt; text-decoration:none }
#pre a:hover { color:red; background-color:#6699CC; border:#006699
2px solid }
#pre a:active { color: white }
#pre a:visited { text-decoration: underline }
-->
</style>
```

Descendant selector

- Descendant به این معنی است که یک عنصر، فرزند عنصر دیگر باشد. سبک استفاده شده برای یک عنصر پدر بر روی تمامی عناصر فرزند اعمال می شود. مثال:

```
<div class="a1" id="a2">  
  <p>  
  This is some text in paragraph  
  </p>  
  <table> ..... </table>  
  <a href= "#"> ..... </a>  
</div>
```

Descendant selector....

• نکته:

```
div.b1 h2
{
Font-size:18px;
Margin-top: 0;
}
```

سبک فوق بر روی عنصر h2 ای که داخل یک تگ div با کلاس b1 قرار دارد اعمال می شود.

شناور ساختن عناصر و تراز عمودی

خاصیت float

- از این خاصیت برای شناور کردن عناصر و قرار دادن آن ها در کنار یکدیگر استفاده می شود و یکی از سه مقدار را می پذیرد:
- None: مقدار پیشفرض (هیچ)
- left: عنصری که این مقدار برای آن تخصیص یابد در سمت چپ صفحه و یا در سمت چپ عنصر پدر خود قرار خواهد گرفت و عناصر پس از آن نیز در کنار این عنصر شناور خواهند شد.
- Right: عنصری که این مقدار برای آن تخصیص یابد در سمت راست صفحه و یا در سمت راست عنصر پدر خود قرار خواهد گرفت و عناصر پس از آن نیز در کنار این عنصر شناور خواهند شد.
- **خاصیت float برای ایجاد و تنظیم بخش navigation (منوی سایت) مناسب است.**

شناور ساختن عناصر و تراز عمودی...

- مثال: ایجاد بخش منوی بالای سایت
(کد htm)

```
<h1> MyWebsite.com </h1>  
<ul class="navi">  
<li> <a href="#"> Home Page </a> </li>  
<li> <a href="#"> Services </a> </li>  
<li> <a href="#"> About Us </a> </li>  
</ul>
```

شناور ساختن عناصر و تراز عمودی...

- مثال: ایجاد بخش منوی بالای سایت
(کد CSS)

```
h1 {  
  Float: left;  
  Margin-top: 0;  
}
```

```
.navi {  
  Float: right;  
  Margin: 0;  
  List-style: none;  
}
```

```
.navi li {  
  Float: left;  
}
```

```
.navi a {  
  Display: block;  
  Margin-left: 15px;  
  Padding: 15px;  
  Border: 1px solid #ccc;  
  Color: red;  
  Text-decoration: none;  
}
```

```
.navi a:focus , .navi a:hover {  
  Background: #aaa;  
  Color: #fff;  
}
```

شناور ساختن عناصر و تراز عمودی...

خاصیت Clear

به منظور از بین بردن تاثیر عنصر floated بر روی عناصر همجواری از خاصیت clear استفاده می کنند. یکی از مقادیر زیر را می گیرد:

- مقدار پیشفرض : none
- left : تاثیر خاصیت float:left را از بین می برد؛ یعنی باعث می شود که تاثیر عنصری که خاصیت float left به آن دادیم بر روی عناصر دیگر از بین برود.
- right : تاثیر خاصیت float:right را از بین می برد.
- both : تاثیر هر دو خاصیت float:left و float:right را بین می برد.

اگر خاصیت clear برای عنصری تعریف شود این عنصر در کنار عنصر floated شناور نخواهد شد.

شناور ساختن عناصر و تراز عمودی...

- **مثال:** فرض کنید در مثال قبلی کد زیر به انتهای کدهای html اضافه شود:

```
<h2> The learning of Css is interesting... </h2>
```

اگر در اینحالت از clear استفاده نکنیم، صفحه بهم ریخته خواهد شد. پس کد زیر را به انتهای کدهای CSS اضافه می کنیم تا مشکل حل شود:

```
H2 {  
Clear: both;  
}
```

شناور ساختن عناصر و تراز عمودی...

• خاصیت vertical-align

این خاصیت به منظور تراز بندی عمودی عناصر inline مثل `` , `` , `<a>` و ... استفاده می شود.

تفاوت عناصر inline و عناصر block:

- عناصر block عناصری هستند که قبل و بعد از آنها یک خط خالی وجود دارد؛ مانند `<h1>` , `<div>` , `<p>` و ...
- عناصر inline عناصری هستند که قبل و بعد از آنها خط خالی وجود ندارد؛ مانند `<a>` , `` , `` و ...

شناور ساختن عناصر و تراز عمودی...

خاصیت **vertical-align** یکی از مقادیر عددی، درصدی یا کلمات زیر را می پذیرد:

- **baseline** : این مقدار خطی است فرضی که هم تراز با قسمت پایینی حروف یک کلمه یا جمله می باشد.
- **middle** : تراز با میانه ی خط
- **super**: تراز بالای خط مثل اعداد توان در ریاضی؛ این مقدار معادل تگ `<sup>` در HTML است.
- **sub**: تراز با پایین خط و مثل اعداد اندیس در ریاضی ؛ این مقدار معادل تگ `<sub>` در HTML است.
- **Text-top** : متن را با بالای بلندترین حروف اطراف تراز می کند.
- **Text-bottom** : متن را با کوتاه ترین یا پایین ترین حروف تراز می کند،
- **top** : قسمت بالای عنصر را با بلند ترین عنصر موجود در خط تراز می کند.
- **bottom** : قسمت پایین عنصر را با کوتاه ترین عنصر موجود در خط تراز می کند.

شناور ساختن عناصر و تراز عمودی...

• **نکته 1:** استفاده از مقادیر درصدی و عددی بستگی به خاصیت `line-height` تعریف شده برای عنصر مورد نظر دارد. مثلاً استفاده از `vertical-align: 50%` متن را در وسط خط تراز می کند.

• **نکته 2:** خاصیت `vertical-align` به طور خاص برای تراز بندی تصاویر مفید است.

شناور ساختن عناصر و تراز عمودی...

مثال: (HTML)

• مثال: (css)

```
<h1>  
  <span class="top"> Top </span>  
  <span class="middle"> Middle  
</span> CSS learning  
  <span class="bottom"> bottom  
</span> CSS learning  
</h1>
```

```
H1 span { font-size: 10px;}  
.top {vertical-align: top;}  
.bottom {vertical-align: bottom;}  
.middle {vertical-align: middle;}
```


لیست ها در CSS

- لیست ها عناصری با کاربرد های بسیار متنوع هستند. از جمله برای بخش navigation سایت و برای ایجاد کنترل های tab استفاده می شوند. Navigation معمولاً شامل لیستی از لینک ها است، که دارای ویژگی های بصری می باشد و ثابت و خشک نیست. در این بخش به طریقه استایل دادن و معرفی خصوصیت هایی که برای کار با لیست ها در CSS استفاده می شود می پردازیم.

لیست ها در CSS...

- با استفاده از این خاصیت می توان برای لیست های مرتب و نامرتب bullet تعریف کرد. و یکی از 14 مقدار زیر را می پذیرد.

disc , circle , square , decimal , decimal-leading-zero , lower-roman , upper-roman , lower-greek , lower-latin , upper-latin , armenian , georgian , none

- مقدار پیشفرض برای لیست های نامرتب (ul) : disc
 - مقدار پیشفرض برای لیست های مرتب (ol) : decimal
- Bullet های مخصوص لیست های نامرتب عبارتند از : square ، circle و disc و none و بقیه bullet های مربوط به لیست های مرتب می باشد.**

لیست ها در CSS...

مثال:

```
<h2> Web Design Learning</h2>
<ul>
  <li>HTML
    <ol class="html">
      <li>HTML 1</li>
      <li>HTML 2</li>
      <li>HTML 3</li>
    </ol>
  </li>
  <li>CSS
    <ol class="css">
      <li>CSS 1</li>
```

```
      <li>CSS 2</li>
      <li>CSS 3</li>
    </ol>
  </li>
  <li>PHP</li>
  <li>ASP</li>
</ul>
```

و کد CSS:

```
ul li {list-style-type: square;}
ul ol.html li {list-style-type: lower-alpha;}
ul ol.css li {list-style-type: upper-roman;}
```

لیست ها در CSS...

• خاصیت `list-style-image`

این خاصیت یک عکس را برای استفاده به عنوان نشانه `bullet` در کنار عنصر لیست تعریف می کند. این عکس با نشانه ی تعریف شده در خاصیت `list-style-type` جایگزین می شود.

• مثال:

```
ul li {list-style-image: url('images/green-bullet.png');}
```

```
ul ol.html li {list-style-image: url('images/blue-bullet.png');}
```

```
ul ol.css li {list-style-image: url('images/brown-bullet.png');}
```

لیست ها در CSS...

• خاصیت `list-style-position`

این خاصیت موقعیت و نحوه ی قرار گرفتن `bullet` در کنار آیتم های لیست را مشخص می کند. و یکی از دو مقدار زیر را می گیرد:

• `inside` : مشخص می کند که `bullet` لیست باید در داخل آیتم های لیست قرار گیرد.

• `outside` : مشخص که `bullet` لیست باید در خارج از آیتم های لیست قرار گیرد.

• مثال:

```
ul li {list-style-type: square;}
```

```
ul ol.html li {  
  list-style-position: inside;  
  background: #09f; }
```

```
ul ol.css li {  
  list-style-position: outside;  
  background: #ff9; }
```

لیست ها در CSS...

• کوتاه نویسی خاصیت `list-style`

برای کوتاه نویسی خاصیت `list-style` از قاعده زیر استفاده می شود:

```
list-style: type position url('ImageAddress');
```

مثال:

```
li { list-style: square inside url('bullet1.png'); }
```

موقعیت عناصر در صفحه

- Positioning به شما این امکان را می دهد که موقعیت هر عنصر را به دقت در یک سند مشخص کنید.
- می توان مشخص کرد که موقعیت یک عنصر به عنصر پدر خود وابسته باشد یا به پنجره ی مرورگر.
- با استفاده از positioning می توان یک عنصر را بر روی عنصر دیگر لایه بندی کرد.
- Positioning شامل خاصیت position و چهار مقدار left , right , bottom ,top می باشد که مختصات دقیق محل قرار گیری عنصر را میتوان با استفاده از آن ها تعیین کرد.

موقعیت عناصر در صفحه...

• خاصیت position

این خاصیت به تنهایی کار خاصی را روی عنصر انجام نمی دهد و همیشه به همراه خاصیت های left, right, bottom یا top می آید، اما قبل از مقدار دهی به این خاصیت ها در اول حتما باید به خصوصیت position یکی از چهار مقدار زیر را داد:

- **Static**: مقدار پیشفرض، و همان حالت عادی قرار گرفتن عناصر طبق ترتیبی که در سند HTML است می باشد.
- **Relative**: اجازه می دهد که یک عنصر در یک محل خاص از سند قرار گیرد، و وابسته به عنصر پدر می باشد.
- **Absolute**: اجازه می دهد که یک عنصر در یک محل خاص از سند قرار گیرد، و وابسته به پنجره مرورگر می باشد.
- **Fixed**: عناصر fixed همواره در محل خود باقی می مانند حتی اگر سند scroll شود.

موقعیت عناصر در صفحه...

• مقدار Static

مقدار static همانطور که گفتیم تاثیری بر روی عنصر ندارد و همان حالت default عناصر می باشد، بجز در عناصر لیست ها (li, ol, ul).

مقدار پیشفرض برای عناصر لیست ها none است، اما اگر برای عناصر لیستها از مقدار static استفاده کنیم حالت تو در توی آن ها از بین خواهد رفت.

نکته:

عناصری که مقدار absolute، relative یا fixed دارند را می توان با استفاده از مقادیر left، right، top و bottom تعیین موقعیت کرد.

موقعیت عناصر در صفحه...

• مقدار Relative

این مقدار موقعیت یک عنصر نسبت به عنصر پدر را در صفحه تعیین می کند و نقطه ارجاع (نقطه شروع حرکت عنصر) از مختصاتی که عنصر پدر دارد شروع خواهد شد. یا به عبارتی دیگر عنصر از محلی که در آن قرار دارد شروع به مختصات گیری می کند. مثال (بدون موقعیت دهی) -

```
<div class="father">  
  <div class="children"> </div>  
</div>
```

کد html:

موقعیت عناصر در صفحه...

و کد CSS:

```
* {  
  margin: 0;  
  padding: 0;  
}  
  
body {  
  width: 500px;  
  margin: 10px auto;  
}
```

```
div.father {  
  width: 400px;  
  height: 300px;  
  margin: 0 auto;  
  border: 4px solid green;  
}  
  
div.children {  
  width: 150px;  
  height: 150px;  
  background: #09f;  
  border: 3px solid red;
```

موقعیت عناصر در صفحه...

- حال سبک عنصر فرزند را در مثال فوق تغییر می دهیم (با موقعیت دهی):

```
div.children {  
  width: 150px;  
  height: 150px;  
  background: #09f;  
  border: 3px solid red;  
  position: relative;  
  top: 10px;  
  right: 20px;  
}
```

نکته: استفاده از مقادیر منفی
برای خصوصیات top و bottom مجاز می باشد.

موقعیت عناصر در صفحه...

• مقدار absolute

این مقدار موقعیت یک عنصر را نسبت به پنجره ی مرورگر تعیین می کند یعنی نقطه شروع عنصر از یکی از گوشه های پنجره (با توجه به خصوصیتی که تعیین خواهیم کرد) خواهد بود.

مثال:

```
<div class="father">  
  <div class="children"> </div>  
</div>
```

کد html:

موقعیت عناصر در صفحه...

و کد CSS:

```
* {  
  margin: 0;  
  padding: 0;  
}  
body {  
  width: 500px;  
  margin: 10px auto;  
}
```

```
div.father {  
  width: 400px;  
  height: 300px;  
  margin: 0 auto;  
  border: 4px solid green; }  
div.children {  
  width: 150px;  
  height: 150px;  
  background: #09f;  
  border: 3px solid red;  
  position: absolute;  
  top: 10px;  
  right: 20px; }
```

موقعیت عناصر در صفحه...

• مقدار **fixed**

این مقدار موقعیت عنصر را همانند مقدار **absolute** وابسته به صفحه مرورگر خواهد کرد، یعنی نقطه شروع حرکت عنصر از یکی از گوشه های پنجره مرورگر خواهد بود با این تفاوت که عنصری با مقدار **fixed** پس از موقعیت دهی به آن همواره در محل خود باقی می ماند، حتی اگر صفحه **scroll** شود.

موقعیت عناصر در صفحه...

• مثال برای **fixed**

```
<html>
<body>
<div style="position:fixed;top:50px;right:100px;border-style:double"> This text is fixed.</div>
<p> Alireza Jomeiri<p></br>
<p> Alireza Jomeiri<p></br>
<p> Alireza Jomeiri<p></br>
<p> Alireza Jomeiri<p></br>
<p> Alireza Jomeiri<p></br>
<p> Alireza Jomeiri<p></br>
<p> Alireza Jomeiri<p></br>
<p> Alireza Jomeiri<p></br>
</body>
</html>
```


طریقه نمایش و لایه بندی عناصر

- خصوصیت display

با استفاده از این خصوصیت می توان طریقه نمایش عناصر را تعیین کرد و یکی از 22 مقدار زیر را می گیرد:

none , block , inline , inline-block , list-item , run-in , compact ,
table , table-row-group , table-header-group , table-footer-group ,
table-row , table-column-group , table-column , table-cell , table
caption , ruby , ruby-base , ruby-text , ruby-base-group , ruby-
text-group

طریقه نمایش و لایه بندی عناصر...

• پر کاربرد ترین مقادیر در بین این 22 مقدار :

- None: عدم نمایش عنصر (مخفی کردن)
- Block: نمایش عنصر (اگر مخفی بود) و به این صورت که یک خط خالی قبل و بعد از خود ایجاد خواهد کرد.
- Inline: اگر این مقدار را به یک عنصر block بدهیم، خط قبل و بعد از خود را از بین میبرد.
- List-item: عنصر را به شکل لیست درمی آورد (مانند استفاده از تگ li)

طریقه نمایش و لایه بندی عناصر...

- خاصیت z-index

خاصیت z-index موقعیت عناصر را در طول محور نامرئی z کنترل می کند. مشروط به اینکه عناصر به صورت absolute , relative یا fixed موقعیت دهی شده باشند.

قاعده کلی:

z-index: auto | integer

طریقه نمایش و لایه بندی عناصر...

- در حالت auto اگر چند عنصر با یکدیگر تداخل داشته باشند، عنصری که آخر از همه در ترتیب عناصر فایل HTML وارد شده در صفحه نمایش بالاتر قرار میگیرد (نسبت به محور z که بر صفحه نمایش عمود است) و باقی عناصر نیز به همین ترتیب گفته شده قرار خواهند گرفت (در حالت auto عنصری که اول از همه وارد شود در زیر سایر عناصر قرار خواهد گرفت). میتوان با تخصیص مقادیر integer به خاصیت z-index ترتیب قرار گیری عناصر در راستای محور z را تغییر داد. عنصری که مقدار عددی بالاتری دارد بالا (مافوق) سایر عناصر در صفحه نمایش داده خواهد شد و سایر عناصر نیز به ترتیب بزرگی مقدار خود در صفحه ظاهر خواهند شد.

- نکته : IE6 و IE7 از z-index پشتیبانی می کنند اما با کمی اشکال.

آشنایی با خواص opacity، visibility و cursor

- خصوصیت opacity

با استفاده از خاصیت opacity می توان میزان شفافیت یک عنصر را کنترل کرد.
قاعده کلی:

opacity: <number>

- مقدار number عددی بین صفر و یک می باشد و عددی اعشاری است.
- مقدار 0 تیره ترین حالت و مقدار 1 شفاف ترین حالت برای opacity می باشد.
- مقدار پیش فرض number برابر 1 است.

آشنایی با خواص opacity، visibility و cursor...

• مثال: (کد html)

```
<div>  
  <h1> Item1 </h1>  
  <h2> Item2 </h2>  
  <h3> Item3 </h3>  
  <h4> Item4 </h4>  
  <h5> Item5 </h5>  
</div>
```

آشنایی با خواص opacity، visibility و cursor...

• مثال: (کد css)

```
div {background: #000;}
```

```
h1,h2,h3,h4,h5 {
```

```
  color: #000;
```

```
  font-size: 30px;
```

```
  display: inline;
```

```
  padding: 5px 10px;
```

```
  margin: 0 10px;
```

```
  background: red;
```

```
}
```

```
h1 { opacity: .20; }
```

```
h2 { opacity: .40; }
```

```
h3 { opacity: .60; }
```

```
h4 { opacity: .80; }
```

```
h5 { opacity: 1; }
```

آشنایی با خواص opacity، visibility و cursor...

- خصوصیت visibility

با استفاده از این خاصیت می توان یک عنصر را در صفحه نمایش داد و یا مخفی کرد.
قاعده کلی:

Visibility: visible | hidden;

مقدار پیش فرض: visible

آشنایی با خواص opacity، visibility و cursor...

• خصوصیت cursor

با استفاده از خاصیت cursor می توان نوع اشاره گر ماوس مورد استفاده برای عناصر را تغییر داد.

قاعده کلی:

Cursor: url(' ') | auto | crosshair | default | pointer | move | e-resize | ne-resize | nw-resize | n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help | progress

نکته: با استفاده از مقادیر url می توان یک custom cursor را بر اساس یک `img` تعریف کرد. در عمل این روش تقریباً هرگز استفاده نمیشود و برای استفاده توصیه نمی شود و استفاده از `cursor` های استاندارد بهتر می باشد.

جلوه های ویژه در CSS3

- خاصیت transform

با این خاصیت می توان بر روی عناصر جلوه های ویژه اعمال نمود. سه مقدار دارد:

Transform: rotate(numberdeg)

Transform: skew(numberdeg)

Transform: scale(number)

- Rotate: برای چرخش اجزا به کار می رود.
- Skew: برای کجی و انحراف عناصر به کار می رود
- Scale: برای بزرگنمایی عناصر به کار می رود

جلوه های ویژه در CSS3...

- مثال برای transform:

```
<html>
<head>
<style>
img { -webkit-transform: rotate(50deg);
      -moz-transform: rotate(50deg);
      -o-transform: rotate(50deg);
      transform: rotate(50deg); }
</style>
</head>
<body>

</body>
</html>
```

جلوه های ویژه در CSS3...

- مثال برای skew:

```
transform: skew(50deg);
```

- مثال برای scale:

```
transform: scale(1.5);
```

عدد 1.5 بدین معناست که عرض و ارتفاع عنصر 1.5 برابر افزایش می یابد.